PIK, Vol. 13, pp. 261–267, Dezember 2010 · Copyright © by Walter de Gruyter · Berlin · New York. DOI 10.1515/piko.2010.045

BIGML: A Location Model with Individual Waypoint Graphs for Indoor Location-based Services

Moritz Kessel

Mobile and Distributed Systems Group
Institute for Informatics
Ludwig-Maximilians-University
Munich
Oettingenstr. 67
80538 Munich
Germany
moritz.kessel@ifi.lmu.de

Moritz Kessel is a research associate at the Mobile and Distributed Systems Group at LMU Munich. His main research interests include location models for LBS, especially inside of buildings, indoor navigation systems and positioning systems.

Peter Ruppel

Mobile and Distributed Systems Group
Institute for Informatics
Ludwig-Maximilians-University
Munich
Oettingenstr. 67
80538 Munich
Germany
peter.ruppel@ifi.lmu.de

Peter Ruppel is a research associate at the Mobile and Distributed Systems Group at LMU Munich. His main research interests include location-based services, positioning, navigation and mobile communications.

Florian Gschwandtner

Mobile and Distributed Systems Group
Institute for Informatics
Ludwig-Maximilians-University
Munich
Oettingenstr. 67
80538 Munich
Germany
florian.gschwandtner@ifi.lmu.de

Florian Gschwandtner is a research associate at the Mobile and Distributed Systems Group at LMU Munich. His research interests include location-based services, both in indoor and outdoor environments and navigation in indoor environments.

Abstract

Indoor Location-based Services (I-LBS) provide information that is compiled by taking into account the current locations of targets as well as models of surrounding buildings. Beside navigation applications there are lots of other examples on the field of workflow optimization or logistics. A key role among the value chain of I-LBS is a location model provider that delivers topological and geometric information about a building. This includes for example computing the walking distance between two locations, finding certain points of interest or looking up the room name for a target's coordinate, which has been measured by an arbitrary indoor positioning system. Building models have already been studied in the past. However, existing approaches focus either on the visualization of a building or they are intended to support architects in the process of constructing the building. In this paper we identify requirements for future I-LBS and we present a new location model that allows to store both topological and geometric properties of buildings. The model is specified as an application schema for the Geography Markup Language (GML). It enables a location model provider to utilize automatically generated waypoint graphs that can be individually adapted to the requirements of the specific I-LBS. Further more the waypoint graphs can easily be replaced leaving the remaining geometrical and topological properties unchanged. The feasibility of the model for I-LBS as well as different approaches for automated model generation are also demonstrated by a working prototype.

1 Introduction

The exploding number of smartphones as well as various marketplaces for mobile applications have greatly favored the growing popularity of mobile services. Together with accurate outdoor positioning technologies such as GPS, Location-based Services (LBS) have entered the mass market. At the same time so-called Indoor Location-based Services are emerging. However, the spread of I-LBS is currently limited mainly by two factors: First, indoor positioning systems are either expensive or they lack the needed accuracy. Second, the lack of comprehensive indoor location models makes it difficult to offer attractive services to the user.

Location models for outdoor environments have been examined in detail in the past years, especially the Open Geospatial Consortium (OGC) has published standards for location modeling such as KML [17] and CityGML [18] and has as well standardized the creation of application schemata for location models with GML [16]. Another example is given by the Openstreetmap [8] project, which has grown extensively in the recent past and offers a flexible approach for modeling semantically enriched maps.

I-LBS require not only accurate indoor positioning technologies, but also location models that reflect a building's general structure and topology. The de facto standard for exchanging architectural plans, AutoDesk DXF [11], was not developed to contain the semantic information needed by most I-LBS. Existing standards for indoor location models such as IFC [6] and CityGML offer various options to describe the properties of buildings. However, these standards focus either on the visualization of building information or they are intended to be used mainly by architects. Future I-LBS will require additional information, especially the accessibility and general topology of rooms, floors and other parts of a building. Such information can be expressed for example by using waypoint graphs and the geometric shape of objects. Among the value chain of I-LBS (compare [20]) this task is accomplished by the role of the location model provider, which encapsulates the location model and provides interfaces to the I-LBS provider.

Given that different I-LBS require various levels of detail of the building, the location model provider should also be able to deliver topological information with a varying granularity and accuracy. An example could be that a simple walking distance calculator will only need a very coarse waypoint graph, whereas an augmented navigation application will greatly benefit from a high resolution path through the building.

Existing location models for I-LBS do not allow an easy exchange of walkability information and they are not oriented towards automated waypoint graph generation so far.

In this paper we present BIGML, a new location model that takes into account the special requirements of I-LBS. It is specified as an application schema for GML 3.2 and it allows to separate the waypoint graph and its granularity from the rest of the geometric and topological aspects. This enables the exchange of the underlying graph structure by choosing different algorithms for automatic waypoint graph creation. Being outside the scope of this paper, the automated waypoint graph generation has been successfully demonstrated with various algorithms including triangulation, convex partitioning, corner graphs and straight skeleton [19].

BIGML also allows to explicitly describe points of interest (POI) as well as areas of interest (AOI). The latter is a concept to group areas outside the strict hierarchy of rooms and floors. Furthermore, the model can be created automatically from existing CAD data, which is demonstrated by a prototype that parses DXF files.

The remainder of this paper is structured as follows: In the following section requirements for a location model for future I-LBS are identified and discussed. Section 3 introduces the location model, describes the contained elements in detail and elaborates on automated waypoint graph creation. Section 4 shows the results of an empirical evaluation. Section 5 covers related work and Section 6 concludes the paper with some remarks on future work.

2 Location Models for I-LBS

Location models for I-LBS have several requirements that are different from outdoor LBS models. These result from the roles and actors involved as well as from available indoor positioning technologies and the class of service being provided. Given that already today there is a variety of I-LBS available and running, each service has slightly different requirements with respect to the location model. To meet these requirements, we analyzed several existing I-LBS of varying classes, including information, navigation, tracking and management services [10, 15, 23, 24], adapting the classification from [20] for indoor environments.

A target's position, which has been measured by a positioning system, needs to be mapped onto the location model and vice versa. The location model itself can be expressed either in a symbolic, geometric or hybrid fashion [12]. Symbolic models involve the assignment of human-readable designators to locations and they can be further subdivided into set-based and graph-based approaches. The geometric models use spatial coordinate systems to store location data and are essential for computations such as distance between coordinates or rendering of maps. A hybrid location model comprises both symbolic and geometric features. Given that many positioning systems deliver coordinates on a local scale it is desirable for a location model to allow easy transformation between model-specific and other coordinate systems.

A second fundamental requirement is that routes can be computed. Optimal paths based on the walking distance are required not only by indoor navigation applications, but also for example for logistics, maintenance scheduling or workflow optimization. Other applications may also require constraints to be taken into account such as the maximal width along a path or a turn radius. A waypoint graph can be used to determine a desired route. In most cases the waypoints of the graph will also be linked to geometric coordinates in order to connect topological with geometrical information. However, the required resolution and accuracy of waypoints may vary from application to application. Thus the creation of a graph should not affect the building's structure and topology.

Querying *k* nearest objects of interest is another mechanism required by many I-LBS. The objects can be sorted with respect to the walking distance, the Euclidean distance or depending on other attributes. Also the query can refer to static or dynamic targets. Depending on the mobility of the target, this information is either stored in the static location model or is directly kept in the applications memory.

In contrast to the nearest neighbor query, a range query queries for all objects of a certain type in a fixed geographic area. To support range queries, a location model has to model object positions. Containment of symbolic coordinates should be explicit by construction, e.g. a building should contain its floors which in turn contain their rooms, while the containment of geometric coordinates in a given area can be tested on demand.

When it comes to the visual presentation and map rendering for a location model it is vital to have geometric information about the locations and orientations of rooms, floors and buildings available.

In addition to the fundamental shape of a building and its rooms, it is also necessary to describe POIs, which can be related to a certain object or asset. Besides single points it is also desirable to model whole AOIs at which the covered area does not necessarily correspond to the topology of the waypoint graph (meaning that an AOI can cover e.g. two subareas in adjacent rooms which are separated by a wall and thus two points in the AOI might not be directly reachable without leaving the AOI).

The diversity of I-LBS is the reason for another requirement for a comprehensive location model: In order to be able to model every possible feature of an object that can be of interest to a service, a flexible way to store the properties is required. Free tagging describes the possibility to assign any number of additional attributes to existing objects.

Existing location models achieve already various of these requirements. However, an explicit and flexible representation for waypoint graphs is missing so far, especially one that allows an easy and automated graph generation based on varying waypoint graph generation algorithms.

3 BIGML: A Location Model for I-LBS

In the following, we present BIGML[1], an application schema for GML. GML already offers a huge variety of types and elements for the presentation of geometric, topological and temporal information [16]. It serves as a construction kit for application specific schemas. A list of existing application schemas such as the Canadian Road Markup Language, the Climate Science Modeling Language or XPlanGML, can be found at [2].

In this section the relevant elements and corresponding concepts are introduced. Figure 1 shows a UML diagram and gives an overview on the classes, associations and inherited classes involved. At the end of the section the integration of different automatically generated waypoint graphs is discussed.

3.1 Location Model Elements

GML offers a wide range of so called data types, which are specifications of value domains. We introduced a data type named **PropertyTag** which is a key-value pair fulfilling the requirements of free tagging. It enables the model provider to store additional content and information in the model. The two attributes of a *PropertyTag* element, *key* and *value*, are strings with any content.

The class **GlobalCRSReferencePoint** inherits from *AbstractGeometry*. Each *GlobalCRSReferencePoint* consists of two *gml::Point* objects, one with coordinates in the local system and the other with coordinates in a wellknown system such as WGS84. While the topic of coordinate transformation and map projection is outside the scope of this paper, three common points in two coordinate systems (CS) are sufficient to compute the transformation parameters. Those parameters are required to map a point from one CS into the other and vice versa. No explicit transformation is given, because the model should be as simple as possible, enabling the model creator to choose the reference points manually. The local coordinate system is given as a child element of the site, which is described later on.

Features are represented by the abstract superclass *AbstractBuildingInformation*, which is derived from the GML-class *AbstractFeature*. An **AbstractBuildingInformation** element has an arbitrary number of *PropertyTags* as described above. The subclasses represent abstractions of the real world phenomena [13] such as rooms, doors, entries or POIs. Direct subclasses are listed in the following paragraphs.

A **Site** represents a site or real estate, which can hold one or more *buildings*, several *entries* of type *SiteEntry* and a reference to all *BuildingPortals*, connecting the inside of buildings with the outside world on the site. Furthermore each *Site* has a local coordinate system *localCS*, which is the reference system for all geometric coordinates of the model (except for the *globalPoint* children of any *globalCRSReferencePoint*). The geometric representation of the *Site* is given by a *gml::Polygon* in the *outline* child of the site. With at least three optional *globalCRSReferencePoints* a transformation between two reference systems becomes possible. A *Site* has the optional string-attributes *class* and *usage*, which describe the kind of site and the actual usage of the site. The mechanism is adopted from CityGML where such attributes are given for buildings and rooms [18, Annex C]. An optional *owner* and any number of associated *websites* can be given. Only the globally unique *URI* is mandatory and should be used to identify the site. Last but not least every site has an *graph* object containing the waypoint graph that can be replaced as described in 3.2.

A **Building** represents a building or building part that holds one or more *floors* and belongs to exactly one *site*. It is also linked to a *gml::Polygon* representing its *outline* and the optional attributes *class* and *usage*, whose domain is specified by the German authoritative standards ALKIS/ATKIS [9]. In addition, the number of levels above (*storeysAboveGround*) and below the surface (*storeysBelowGround*), the *height* and *depth* of the building, the *heightAboveNN* (height of ground floor above sea level) and a *yearOfConstruction* can be given. All length attributes such as *height* are of the *gml::LengthType*, which gives the possibility to define the unit of measure.

A **Floor** represents a floor or structural level of a building, containing at least one *room* and belonging to exactly one *building*. It has a *gml::Polygon* as outline and the required attributes *heightAboveGround*, which describes the height of the floor above ground level, which we define as the lowest level above ground, and *level*, which is the integer number of the floors level. The level attribute also defines the vertical order of the floors.

A **Room** represents a single room, containing any number of *POIs*, *AOIs*, *portals* and *nodes*, referencing to the waypoint graph. However, this does not conflict with the ability to replace the waypoint graph as we will describe in 3.2. Every *room* has a *gml::Polygon* as *outline* and references the *floor* it is on. A *Room* has the optional attributes *class* and *usage*, whose domain is specified by the SIG 3D. The optional *heightOfCeiling* parameter is used to create simple 3D representations of the room by combining the outline as 2D polygon with a third dimension. However, this mechanism is not sufficient to model complicated structured ceilings, like roof pitches or galleries. The latter information is often not needed by I-LBS and seldom included in available CAD data. 3D shapes would also result in slower computations, be more expansive in size and have a serious impact on human readability.

[1] The full XML-schema file can be downloaded from: `http://www.mobile.ifi.lmu.de/schemas/bigml`

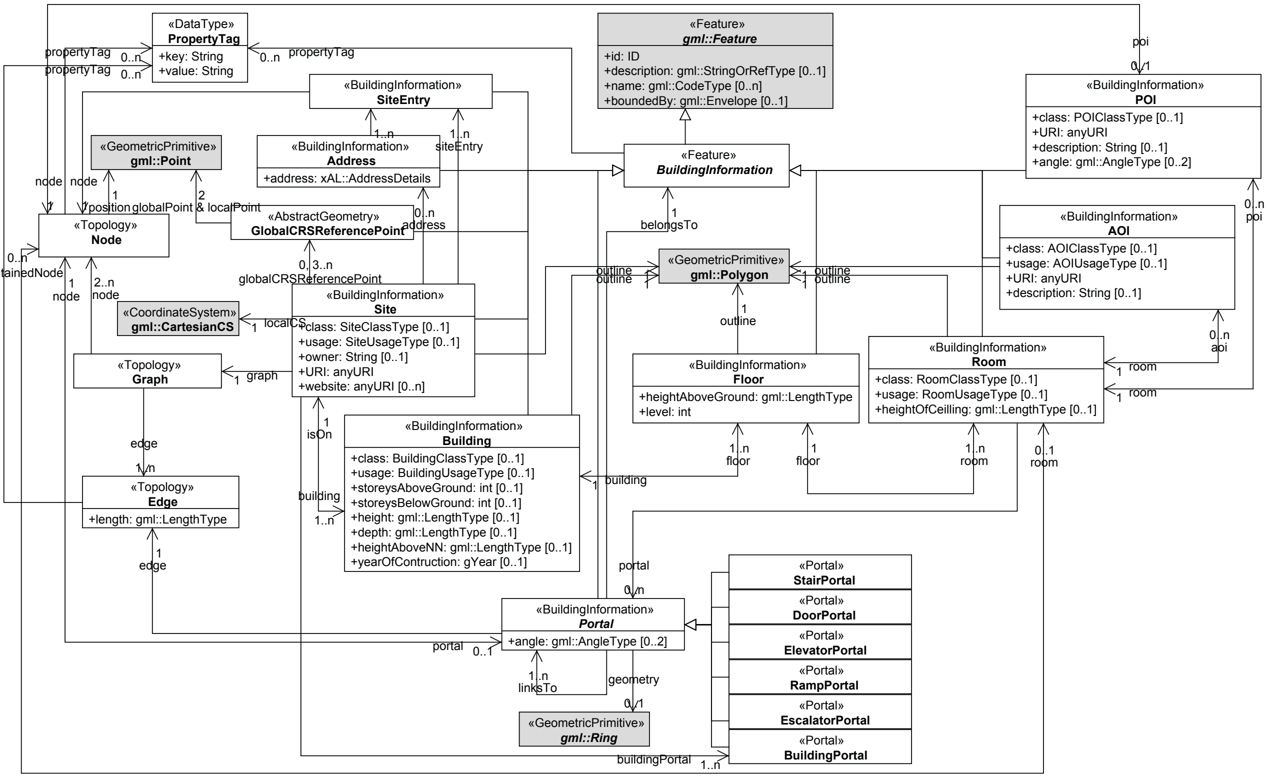

Figure 1 BIGML UML-diagram. Inherited classes from GML 3.2 are shown in gray.

A **POI** represents a single point or object of interest. It references a *node* in the graph, containing the exact topological and geometrical position of the *POI*. Therefore this node should not be overridden when an new graph is assigned to the model. In addition, a *POI* references the *room* it is in and has a unique *URI* and the optional attributes *class*, *description* and two *angles* (measured anti-clockwise from the first axis of the coordinate system), which define a sector within which the POI is visible. This attribute is needed by several I-LBS that utilize line-of-sight.

The class **AOI** extends the mechanism of POI to whole areas of interest. An AOI is not simply a POI with a geometrical extension, but a new powerful concept to define parallel hierarchies. The *URI* is no longer unique, but gives the possibility to aggregate several *AOIs* to a larger area, which can be outside the hierarchy of buildings, floors and rooms. In this context, all *AOIs* with the same URI are seen as a part of the larger area of interest, which is their union. This mechanism can be used to combine several rooms on different floors or even in different buildings, for example in order to model departments in a company or wards in a hospital. Figure 2 shows an example how areas of interest can be used to group rooms. Every *AOI* element references the containing *room*, has a *gml::Polygon* as *outline* and has the optional attributes *class*, *usage* and *description*.

A **Portal** represents a connection between rooms or a site and a room. Each portal belongs to a single *room* or *site* and is linked to at least one other *portal* of the same type. An *AbstractPortal* element can be instantiated by a *DoorPortal*, *StairPortal*, *ElevatorPortal*, *EscalatorPortal*, *RampPortal* or a *BuildingPortal*. Every *Portal* references an *edge* and a *node*. These edges and nodes cannot be replaced together with the waypoint graph. The reason is, that the *Portals* also remain fixed, because they belong to the geometrical or structural representation of the building. As described in 3.2, the nodes and edges referenced by portals are needed to model the walkability between rooms, floors and buildings. In addition the *geometry* of the portal is stored in a *gml::Ring* object, while the viewing angle of the portal entry is modeled by two *angles* of *gml::AngleType*. For these angles the same convention is used as with POIs.

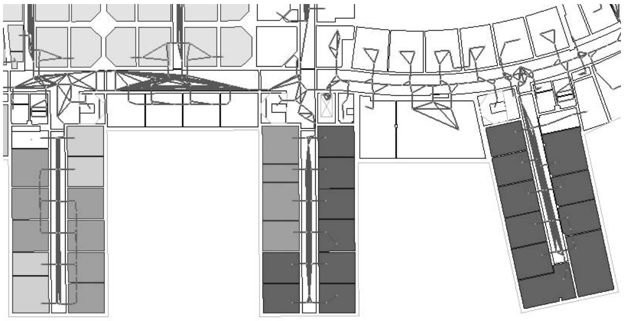

Figure 2 Several areas (different scales of gray) and an automatically created waypoint graph (dark grey edges) in a university building.

A **SiteEntry** represents an entry point to the site from the outside world. It references a single *node* that is part of the topological model of the site. This node is also fixed and not exchangeable.

An **Address** contains the representation of an address in the extensible address language [4] and references a *SiteEntry*, which the address belongs to. It can be seen as a doorplate at the referenced entry, which can include additional information within the property tags.

The following types do not inherit from *AbstractBuildingInformation*, but extend the GML type *AbstractTopology*.

The class **Graph** contains a list of all *nodes* and *edges* of the waypoint graph. There is only one graph object for each site, which should be connected to ensure the reachability of all nodes.

GML offers already a topological type for node elements. It offers many possibilities like pointers to other topological structures containing the node, e.g. faces or topological solids. Since we only need the reference to edges and some building information elements, we define a new type **Node** which references at least one *edge*, exactly one *position* (a *gml::Point* in the local coordinate system) and optionally a single *POI*, *room*, *portal* or *SiteEntry*. Because *Node* does not inherit from *AbstractBuildingInfromation* it has explicitly any number of additional *PropertyTag* elements as children.

The class **Edge** is out of the same reasons not an extension of the GML type edge. The edge defined here is just a simple connection between nodes with a weight and arbitrary additional properties. Every *Edge* references exactly two *nodes*, a *length* of *gml::LengthType* and any number of *PropertyTags*.

3.2 Individual Waypoint Graphs

BIGML allows arbitrary waypoint graph generation approaches to be applied. The possibility to exchange graphs offers a range of opportunities, e.g. the on-demand creation of parts of the graph or offering waypoint graphs with varying granularity for different I-LBS. In the following paragraphs, the stepwise replacement of the graph is described.

We require a predefined set S of nodes and edges which we call static, representing the topological structure (connections between rooms, floors, etc.) of the site. It should stay the same for any underlying waypoint graph. The same applies to the points of interest, which should still be reachable within the new waypoint graph. This set is easily identified: For every edge e which is referenced by a portal and for every node n that references to a portal, site entry or POI the element is added to the set. The remaining nodes and edges are deleted.

In the next step the graph is computed for every room and every part of the site that is not covered by buildings. The algorithm creating the graph can have a large impact on the number of nodes, the length of ways and even the simplicity of ways. Figure 3 shows for example the resulting waypoint graph for a specific room using four different algorithms. An evaluation and discussion on automated waypoint graph generation for I-LBS can be found for instance in [19].

The resulting waypoint graph for each room and the site is connected to the corresponding nodes in S. Different algorithms can be used for the connection, such as connecting the static nodes with all new nodes in line of sight or only with the next new node. Either way it must be ensured that every static node of every room has at least one connection to the new graph of each room.

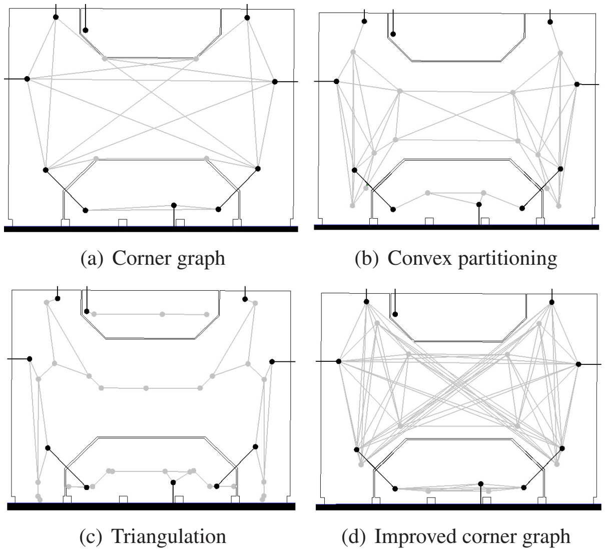
(a) Corner graph
(b) Convex partitioning
(c) Triangulation
(d) Improved corner graph

Figure 3 Waypoint graphs resulting from different algorithms. Black nodes and edges: doors. Gray nodes and edges: exchangeable graph elements.

4 Evaluation

The presented model offers three main advantages: first, it can be integrated into a variety of existing tools for automatic code generation. That allows an easy adoption and instantiation of the model. Second, the geometric and topological elements of the model are independent from each other and can be handled by appropriate algorithms. Given that the search space representation is not restricted by the model, the application developer can e.g. deploy her own R-Trees or A* heuristics. The proposed graph structure of the model also allows hierarchical graphs to be applied. Third, the model supports various types of walkability information. The simplest way is to utilize only existing polygon points and to create e.g. a corner graph. Also the straight skeleton or the medial axis of polygons can be computed and utilized for routing. Furthermore the generation of triangulations, Delaunay triangulations and especially constrained Delaunay triangulations deliver even more information: such navigation mesh can be adopted to calculate shortest paths with a minimal width or for objects with limited turn radii.

The model was created automatically from CAD data, according to the schema specified in Section 3. The CAD data contained a building with five stories, more than 800 rooms and over 1000 portals such as doors, escalators, stairs and elevators (see Figure 4 for the first three sample floors).

Seven different algorithms were successfully used for automated graph generation. A Corner Graph, a variation of

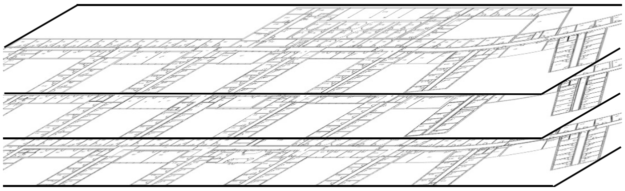

Figure 4 Example: CAD data source for automated location model generation.

the corner graph with waypoints moved to the interior of the polygon, two different algorithms for convex partitioning, a Delaunay triangulation with two different methods for waypoint generation from the resulting triangles and an algorithm to create a straight skeleton were applied. The number of created nodes was in between 2000 and 6500, the number of edges ranged from 5800 to nearly 20000 (depending on the algorithm used). The size of the uncompressed XML file is between 6 MB and 13 MB, mostly influenced by the number of graph elements. Given that the model is rather small in terms of memory space, a complete model can be loaded into the computer's memory for faster access. It is also suitable for deployment on mobile phones.

After generating the model, a map-based routing application and a room management service were implemented to evaluate empirically the applicability of the model. The applications were able to calculate routes and find required rooms as expected. Since the model meets the requirements listed in Section 2, there should hardly be any problems to use the model for other I-LBS.

It has to be stressed that consistency and integrity of the CAD data is of major importance. We tried to parse data from several organizations and buildings, but only very few of them were sufficient to automatically extract a location model without manual modifications.

The model has some limitations, as it is not able to describe complex 3D content. As soon as more 3D I-LBS evolve, the model might need additional information, depending on the required level of detail. So far the model supports only one graph at a time. This is a limitation, because the replacement of the whole waypoint graph is to slow to be carried out on demand. However, it is possible to generate several instances of a building's model, each one with a different waypoint graph.

5 Related Work

The topic of location models has been the focus of extensive research in the past years. There exist three categories of location models: pure indoor location models, outdoor location models and overall location models capable of modeling indoor and outdoor environments.

Existing indoor location models often have their origin in architectural plans, such as CAD or IFC. The de facto standard of architectural CAD from AutoDesk, DXF [11], is used mainly as an exchange format for building plans. It is not easily extensible, because it uses a proprietary data format, and was initially not intended to model topologies. Industrial Foundation Classes (IFC) [5], currently at version 2x3, is a standard of the building industry. It relies on a object oriented model describing all architectural information of buildings. Being a comprehensive and extensive model, IFC also aims mainly for architects and facility management. The same applies for other approaches for standardizing building information, such as the Building Information Spatial Data Model [1] and Green Building XML [3]. Like IFC, these models are far more extensive than BIGML, but are not intended for use on mobile clients in I-LBS. Indoor location models for I-LBS such as the Facility Map Framework [25] often use a proprietary data format not publicly available.

Overall location models often originate from or focus on outdoor location models, since outdoor-LBS is far more common. However, CityGML [18], an application schema for GML, offers a comprehensive approach which has turned recently into a standard of the Open Geospatial Consortium (OGC). It was initially developed to model cities or urban districts in five different levels of detail, ranging from LOD0, a large overview over whole cities, to LOD4 a representation of the inside of buildings. With the LOD4, CityGML offers an indoor location model that is especially suited for visualization, as the geometrical shape of objects is explicitly modeled in 3D. So far it offers no concept to explicitly describe the walkability inside of buildings.

Some research focuses not only on overall location models, but also on platforms to offer and request location information from several sources. Nexus [22] for example offers such a platform which accesses the Nexus Augmented World Model [21]. The latter models many aspects of outdoor and indoor environments, focused on the integration of location-based services.

Several location models such as KML [7] and OpenStreetMap [14] are mainly used for outdoor location modeling, especially for enriching maps and serving for navigation purposes. They fulfill many requirements such as the modeling of POIs and free tagging, may contain a waypoint graph and are already used by mobile clients. However, they offer limited possibilities to describe both structure and topology of buildings in three dimensions.

6 Conclusion and Outlook

In this paper we have presented BIGML, a new location model for Indoor Location-based Services that allows to store both topological and geometric properties of buildings. With BIGML the underlying waypoint graph for a specific building can be individually adjusted for different classes of I-LBS. The model offers a hybrid location model with symbolic and geometric coordinates, it is of hierarchical structure, graph based, extensible and flexible. BIGML is specified as an application schema for GML. For these reasons, the model can be applied directly to different I-LBS with little effort.

Currently we are working on an approach to dynamically fetch parts of the model from applications that run on mobile devices. By transferring only the routes and polygons to the mobile device that are needed to answer the local request of the mobile user, we aim to minimize the amount of transmitted data. Such strategies need to take into account for example the mobility of the target or dynamic changes on the route.

Another open problem remains in the lack of integrity of existing CAD plans. Therefore we are working on a set of algorithms and tools to further support the model creation process. Last but not least, future work will include feasibility studies on utilizing the model to support existing 3D I-LBS such as visually enriched indoor navigation.

References

[1] Building Information Spatial Data Model. `http://bisdm.org`, last visited 07.09.2010.

[2] GML Application Schemas and Profiles. `http://www.ogcnetwork.net/node/210`, last visited 07.09.2010.

[3] Green Building XML. `http://www.gbxml.org`, last visited 07.09.2010.

[4] Extensible Address Language (xAL) Standard Description Document for W3C DTD/Schema. Technical report, Organization for the Advancement of Structured Information Standards, 2002.

[5] IFC2x3 TC1 Documentation. online, June 2010. `http://www.iai-tech.org/ifc/IFC2x3/TC1/html/index.htm`.

[6] ifcXML Schema. online, June 2010. `http://www.iai-tech.org/ifcXML/IFC2x3/FINAL/IFC2X3.xsd`.

[7] KML Reference. online, June 2010. `http://code.google.com/intl/de-DE/apis/kml/documentation/kmlreference.html`.

[8] OpenStreetMap. online, June 2010. `http://wiki.openstreetmap.org/wiki/Main_Page`.

[9] Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland. *Dokumentation zur Modellierung der Geoinformationen des amtlichen Vermessungswesens*, 6.0 edition, April 2008. Available at `http://www.adv-online.de`.

[10] Abhaya Asthana, Mark Cravatts, and Paul Krzyzanowski. An Indoor Wireless System for Personalized Shopping Assistance. Technical report, AT&T Bell Laboratories, 1994.

[11] AutoDesk. *DXF Reference*, 2007.

[12] Christian Becker and Frank Dürr. On Location Models for Ubiquitous Computing. In *Ubiquitous Computing*, volume 9, pages 20–31, 2004.

[13] International Organization for Standardization. ISO 19101 Geographic information – Reference model, 2002.

[14] Mordechai (Muki) Haklay and Patrick Weber. OpenStreetMap: User-Generated Street Maps. *IEEE Pervasive Computing*, 7:12–18, 2008.

[15] Christian Harr. Mobiles Navigationssystem für MedioVis. Master's thesis, Universität Konstanz, 2006.

[16] Open Geospatial Consortium Inc. OpenGIS® Geography Markup Language (GML) Encoding Standard, 08 2007.

[17] Open Geospatial Consortium Inc. Keyhole Markup Language 2.2, 2008.

[18] Open Geospatial Consortium Inc. OpenGIS® City Geography Markup Language (CityGML) Encoding Standard, 08 2008.

[19] Moritz Kessel. Erzeugung von Umgebungsmodellen aus digitalen Gebäudeplänen für Ortsbezogene Dienste. Master's thesis, Ludwig-Maximilians-Universität München, 2010.

[20] Axel Küpper. *Location-Based Services: Fundamentals and Operation*. John Wiley and Sons Ltd., 2005.

[21] Jens Meßmer. Modellierung der Augmented World in Nexus. Master's thesis, Universität Stuttgart, 2001.

[22] Daniela Nicklas. *Ein umfassendes Umgebungsmodell als Integrationsstrategie für ortsbezogene Daten und Dienste*. PhD thesis, Universität Stuttgart, 2005.

[23] Reinhard Oppermann and Marcus Specht. A Context-sensitive Nomadic Information System as an Exhibition Guide. Technical report, German National Research Center for Information Technology – Institute for Applied Information Technology, 2000.

[24] Peter Ruppel, Florian Gschwandtner, Corina Kim Schindhelm, and Claudia Linnhoff-Popien. Indoor Navigation on Distributed Stationary Display Systems. In *33rd Annual IEEE International Computer Software and Applications Conference*, 2009.

[25] Steven A. Shafer. A Framework for Creating and Using Maps of Privately Owned Spaces. In *LoCA*, pages 174–191, 2009.